UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/748,785	12/22/2000	Andrew Cofler	S1022/8583	3330

7590 02/26/2004

James H. Morris
c/o Wolf, Greenfield & Sacks, P.C.
Federal Reserve Plaza
600 Atlantic Avenue
Boston, MA 02210-2211

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	7

DATE MAILED: 02/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/748,785	COFLER ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 December 2000*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *22 December 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *4,5, and 6* .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

1. Claims 1-15 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: #3. Declaration and Fees as received on 3/12/2001, #4. IDS as received on 12/22/2000, #5. IDS as received on 2/12/2002, and #6. IDS as received on 4/14/2003.

Information Disclosure Statement

3. The IDS corresponding to paper #4 (filed on 12/22/2000) lists 5,442,757 (McFarland et al.). However, the McFarland patent has not been provided by applicant. Instead, Pardo et al. (5,754,839) has been provided. It is not clear whether applicant wants the examiner to consider McFarland or Pardo (or both). Consequently, the examiner has not considered either of these documents. In addition, applicant has provided Karp et al., EP 0810519A2, but has not listed it on the IDS. Therefore, Karp has not been considered.

4. The IDS corresponding to paper #6 (filed on 4/14/2003), does not list Nemirovsky (5,857,094), which was provided as a hard copy by applicant. Therefore, Nemirovsky has not been considered.

Specification

5. The disclosure is objected to because of the following informalities:

On page 1, replace "localized" with --localized--.

On page 7, replace "128 bit", "16 bit", and "32 bit" with --128-bit--, --16-bit--, and --32-bit--, respectively.

On page 7, second paragraph, replace "buffer 22. this pair" with --buffer 22. This pair--.

On page 9, the examiner feels it would be more clear if "diagnostic flags diag" is replaced with --diagnostic flags (DIAG)--.

Starting on page 11, replace all occurrences of "synchronisation" (and its variations), with --synchronization--.

On page 12, should "PC register 38" be changed to --PC register 28--?

On page 17, the copending application should be identified by application/patent number and title.

On page 17, last paragraph, it is not clear why the figure is illustrating "four successive clock cycles" when the figure shows seven clock cycles. Are the four cycles those following the program count? Also, the examiner believes that the mentioning of "Figure 7" in the last paragraph may be a mistake.

Appropriate correction is required.

Drawings

6. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: On page 17, last paragraph, numbers 104 and 230 are mentioned but the examiner has been unable to find them in the drawings. A proposed drawing correction or corrected drawings are required in reply to the

Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

7. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference sign(s) not mentioned in the description: In Fig.2, numbers 46 and 47 have not been found in the description. In Fig.7, the field to the right of the "sgls" field has not been mentioned within the description. In Fig.10, number 130 has not been mentioned within the description. A proposed drawing correction, corrected drawings, or amendment to the specification to add the reference sign(s) in the description, are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

8. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference characters "4" and "18" have both been used to designate the processor core. It is not clear from the drawings that number 4 refers to more than just the core. In addition, reference characters "6" and "32" have both been used to designate the OCE PC Watch component. It is not clear from the figure that number 6 refers to more than just this component. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

9. The drawings are objected to because of the following minor informalities: In Fig.2, it looks as if "divert" is misspelled as "divet" near component 47. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

10. Claim 1 is objected to because of the following informalities: Please replace "supplied and from" with --supplied from--. Appropriate correction is required.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 1-5, and 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Song, U.S. Patent No. 5,546,599, in view of Grochowski et al., U.S. Patent No. 6,353,883 (herein referred to as Grochowski).

13. Referring to claim 1, Song has taught a computer system for executing instructions comprising:

a) a fetch unit for fetching instructions to be executed. See Fig.3, and note that instructions are fetched from instruction cache 14 into instruction buffer 70.

b) a decode unit for decoding said instructions. See Fig.3, component 72.

c) at least one pipelined execution unit for executing decoded instructions. From Fig.3, it should be realized that decoded instructions are eventually dispatched to execution units (which are shown in Fig.1). In addition, see column 4, lines 11-16, and note that that the execution units exist within a multi-stage pipeline and that the execution units may require multiple cycles themselves (specifically, in Fig.13 and 15).

d) an emulation unit including control circuitry which cooperates with the decode unit to selectively control the decode unit to implement a precise watch or a non-precise watch on detection of a breakpoint wherein according to a precise watch, the instruction, causing the breakpoint is held at the decode unit and, according to a non-precise watch, the instruction causing the breakpoint and subsequent instructions are permitted to be supplied from the decode unit to the at least one execution unit. Note from the abstract, column 22, lines 30-34, and claim 1 (of Song), that the system is able to operate in precise exception mode. In this mode, the instruction causing the exception (breakpoint) is prevented from being executed (i.e., held at the decode stage). Furthermore, note from column 22, lines 16-29, that the system also operates in a non-precise (imprecise) exception mode. More specifically, the instruction causing the exception (breakpoint) along with subsequent instructions are allowed to proceed through the pipeline so that increased performance is achieved.

e) Song has not taught that:

- the executed instructions are predicated instructions, wherein each instruction includes a guard, the value of which determines whether or not that instruction is executed. However, Grochowski has taught the concept of executing predicated instruction having guards. See Fig.6, step 610, and Fig.1 (note that p2 is a predicate/guard).
- the execution unit is associated with a guard register file holding values of the guards to allow resolution of the guards to be made. However, Grochowski has taught such a concept. See Fig.3A, component 300, and column 2, lines 65-67.

Note that the predicate table (register file), is updated with actual predicate values, which are in turn used as predictions for speculatively executed instructions.

- instructions are supplied to the execution unit while guard resolution in said execution pipeline is awaited. However, Grochowski has taught such a concept. See column 3, lines 4-35. More specifically, predicated instructions are allowed to progress through the pipeline before its corresponding guard is resolved.

A person of ordinary skill in the art would have recognized that by implementing predicated instructions with predicate prediction within Song, a) conditional branches could be eliminated, thereby reducing the amount of instructions required in the instruction set, and b) predicated instructions would be speculatively executed (i.e., executed before the corresponding guard is resolved), thereby maximizing throughput by executing predicated instructions as soon as possible. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Song to include predicated instructions and predicate prediction, as taught by Grochowski.

14. Referring to claim 2, Song in view of Grochowski has taught a computer system as described in claim 1. Song has further taught that the system is implemented on a single chip. See Fig.1, component 10, and column 2, lines 47-48.

15. Referring to claim 3, Song in view of Grochowski has taught a computer system as described in claim 1. Song has further taught a program memory for holding said instructions to be executed. See Fig.1 and column 2, lines 60-61.

16. Referring to claim 4, Song in view of Grochowski has taught a computer system as described in claim 1. Song has not explicitly taught that the emulation unit is associated with an

emulation program memory which holds debug code which is executed in a debug mode. However, recall that Song has taught handling exceptions (see Fig.3 and column 8, lines 10-13). As is known in the art, when an exception/interrupt is triggered during execution of a program, a handling routine must be invoked in order to correct the error associated with the exception/interrupt before execution of the main program may resume. According to "The American Heritage® Dictionary of the English Language, 3rd Edition, 1992," the word "debug" is defined as "to search for and eliminate malfunctioning elements or errors in." Consequently, when an error occurs in an executing program in Song, an exception will be triggered. A handling routine (debug code) is invoked in order to search for the problem causing the exception and eliminate the error. This process would be considered debug mode. In addition, it should be realized that the debug code must be stored somewhere (emulation program memory). As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement an emulation program memory which holds debug code which is executed in a debug mode in Song since this feature is well known, accepted, and expected in the art of exceptions and exception handling.

17. Referring to claim 5, Song in view of Grochowski has taught a computer system as described in claim 1. Song in view of Grochowski has further taught that when the emulation unit is in a precise watch mode, it is operable to issue a request to the execution pipeline for guard resolution, the guard resolution being transmitted to the control circuitry of the emulation unit which is responsive thereto to control operation of the decode unit. Note from column 3, lines 25-35, of Grochowski, that a predicated instruction is prevented from executing until the guard is resolved. This is advantageous in that a time-expensive recovery is not required if the

guard is improperly predicted. Therefore, when Song is in precise watch mode, Grochowski has taught that it is beneficial to issue a request to the execution pipeline for guard resolution for the aforementioned reasons. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to issue a request to the execution pipeline for guard resolution during precise watch mode in Song.

18. Referring to claim 8, Song in view of Grochowski has taught a computer system as described in claim 1. Song has further taught a microinstruction generator which receives instructions from the decode unit and supplies microinstructions to the execution pipeline. See Fig.3, component 74, and column 7, lines 23-26, and note that up to 4 microinstructions that are to be dispatched are determined (aka produced or generated). In addition, Grochowski has taught that said microinstructions include fields for holding respective guards to be resolved. For instance, see column 5, lines 36-41, and note that the predicate is extracted from the instruction. Therefore, there must be a field within the instruction which specifies the predicate.

19. Referring to claim 9, Song in view of Grochowski has taught a computer system as described in claim 1. Song has further taught a plurality of parallel pipelined execution units, including at least two data unit pipelines for executing data processing instructions and at least two address unit pipelines for executing memory access instructions. From Fig.1, it should be seen that floating-point unit (30) and fixed point unit A (22) are data units in that they will execute floating and fixed point instructions (such as add, subtract, mult, etc.), and load/store unit (28) and complex fixed-point unit are address units in that the load/store unit will execute loads and stores, which access memory, and complex fixed point unit also accesses memory (general and special purpose registers, which are forms of memory). Finally, it should be

realized that to make separable is not a generally a patentable feature or would be an obvious improvement. More specifically, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Song's load/store unit into a separate load unit and store unit, thereby yielding two address units.

20. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Song in view of Grochowski, as applied above, and further in view of Matt et al., EP 0943995A2 (as disclosed by applicant and herein referred to as Matt).

21. Referring to claim 6, Song in view of Grochowski has taught a computer system as described in claim 5. Song in view of Grochowski has not taught the specifics set forth in claim 6. However, Matt has taught issuing a go command and divert command to the decode unit responsive to receipt of the guard resolution from the execution pipeline, wherein a go command allows the instruction which caused the breakpoint and subsequent instructions to be normally decoded and executed, and a divert command sets the computer system into a debug mode. See column 4, lines 6-40. More specifically, when a break event occurs, (which would include exceptions taught by Song), the system would issue either a command to resume execution (go command) or a command to execute a service routine (divert command). This allows for real-time execution control for debug functions within a processor, as taught by Matt in column 4, lines 6-9. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Song in view of Grochowski in view of Matt such that an go and divert commands may be issued in order to allow for real-time execution control for debug functions.

22. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Song in view of Grochowski, as applied above, and further in view of Adler et al., U.S. Patent No. 5,627,981 (herein referred to as Adler).

23. Referring to claim 7, Song in view of Grochowski has taught a computer system as described in claim 1. Song in view of Grochowski has not taught the specifics set forth in claim 7. However, Adler has taught a system in which the instruction which caused the breakpoint (exception) and subsequent instructions are decoded and executed normally until such time as said instruction reaches the execution pipeline where its guard is resolved such that a commit signal is generated to the control circuitry of the emulation unit, and wherein the emulation unit is responsive to receipt of the commit signal to set the computer system into a debug mode. See the last 8 lines of the abstract and Fig.9. In essence, predicated instructions are executed speculatively (i.e., executed normally without knowing the actual value of the guard/predicate). When the instruction is at a commit point, and if the instruction's predicate is true, then if that instruction caused a breakpoint (or exception), then it will be serviced (debug mode). This scheme allows errors that should not have occurred (errors resulting from instructions that should not have executed) to be ignored while servicing the errors that should have occurred (errors resulting from instructions that have properly executed. See column 3, lines 20-42. As a result, it would have been obvious to one of ordinary skill in the art to modify the system taught by Song in view of Grochowski such that it includes the functionality of Adler.

24. Claims 10 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Song in view of Grochowski, as applied above, and further in view of Hennessy and Patterson, Computer Architecture - A Quantitative Approach, 2nd Edition, 1996 (herein referred to as Hennessy).

25. Referring to claim 10, Song has taught a method of debugging an on-chip processor which is arranged to execute instructions, the method comprising:

a) fetching instructions to be executed. See Fig.3, and note that instructions are fetched from instruction cache 14 into instruction buffer 70.

b) decoding said instructions. See Fig.3, component 72.

c) executing decoded instructions, said executing step including resolving values of the guards of the instructions. From Fig.3, it should be realized that decoded instructions are eventually dispatched to execution units (which are shown in Fig.1).

d) and detecting instructions which have a debug effect and acting on said instructions in dependence on whether the processor is in a precise watch mode or a non-precise watch mode wherein, according to a non-precise watch mode, the instruction and subsequent instruction are supplied and executed normally. Note from the abstract, column 22, lines 30-34, and claim 1 (of Song), that the system is able to operate in precise exception mode. In this mode, the instruction causing the exception (breakpoint) is prevented from being executed (i.e., held at decode stage). Furthermore, note from column 22, lines 16-29, that the system also operates in a non-precise (imprecise) exception mode. More specifically, the instruction causing the exception (breakpoint) along with subsequent instructions are allowed to proceed through the pipeline so that increased performance is achieved. Song has not explicitly taught that according to a precise watch mode, the instruction is not decoded. However, Hennessy has taught that exceptions

(breakpoints) may occur when an instruction is fetched. See Figure 3.41 on page 184. A person would have realized that if an exception occurs during the fetch, then the fetch has failed and the instruction that is supposed to be fetched will not be decoded. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to not decode an instruction during precise watch mode.

e) Song has also not taught that:

- the executed instructions are predicated instructions, wherein each instruction includes a guard, the value of which determines whether or not that instruction is executed. However, Grochowski has taught the concept of executing predicated instruction having guards. See Fig.6, step 610, and Fig.1 (note that p2 is a predicate/guard).
- Said executing step includes resolving values of the guards of the instructions. However, Grochowski has taught such a concept. See Fig.1. Note that by executing the COMPARE instruction, the guard p2 will be resolved.
- instructions are supplied to the execution unit while guard resolution in said execution pipeline is awaited. However, Grochowski has taught such a concept. See column 3, lines 4-35. More specifically, predicated instructions are allowed to progress through the pipeline before its corresponding guard is resolved.

A person of ordinary skill in the art would have recognized that by implementing predicated instructions with predicate prediction within Song, a) conditional branches could be eliminated, thereby reducing the amount of instructions required in the instruction set, and b) predicated instructions would be speculatively executed (i.e., executed before the corresponding

guard is resolved), thereby maximizing throughput by executing predicated instructions as soon as possible. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Song to include predicated instructions and predicate prediction, as taught by Grochowski.

It should be further realized that according to "The American Heritage® Dictionary of the English Language, 3rd Edition, 1992," the word "debug" is defined as "to search for and eliminate malfunctioning elements or errors in." Consequently, when an error occurs in an executing program in Song, an exception will be triggered. A handling routine (debug code) is invoked in order to search for the problem causing the exception and eliminate the error. This process would be considered debugging.

26. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Song in view of Grochowski in view Hennessy, as applied above, and further in view of Kurakazu, U.S. Patent No. 5,644,703 (as disclosed by applicant and herein referred to as Kurakazu).

27. Referring to claim 11, Song in view of Grochowski in view Hennessy has taught a method as described in claim 10. Song in view of Grochowski in view Hennessy has not explicitly taught that breakpoints are detected at instructions having certain program counts. However, Kurakazu has taught that an address break is well known, accepted, and expected in the art. See column 1, lines 12-20. More specifically, a user is allowed to set a breakpoint at a specified address in order to determine the status of the system at that particular point regardless of the instruction that is at the address. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Song in view of Grochowski in

view of Hennessy such that breakpoints are detected at instructions having certain program counts (addresses).

28. Claims 12 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Song in view of Grochowski in view Hennessy, as applied above, and further in view of Christensen et al., U.S. Patent No. 5,752,013 (as disclosed by applicant and herein referred to as Christensen).

29. Referring to claim 12, Song in view of Grochowski in view Hennessy has taught a method as described in claim 10. Song in view of Grochowski in view Hennessy has not taught that breakpoints are detected at instructions having certain opcodes. However, Christensen has taught such a concept. See column 1, lines 64-67. Note that particular instructions may be used as breakpoints. For instance, maybe the program will break at branch-type instructions (which have branch opcodes). Since Christensen has taught that this type of break is common in the art, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement such a feature in Song in view of Grochowski in view of Hennessy.

30. Referring to claim 14, Song in view of Grochowski in view Hennessy in view of Christensen has taught a method as described in claim 12. Furthermore, recall that Song has taught handling exceptions (see Fig.3 and column 8, lines 10-13). As is known in the art, when an exception/interrupt is triggered during execution of a program, a handling routine must be invoked in order to correct the error associated with the exception/interrupt before execution of the main program may resume. According to "The American Heritage® Dictionary of the English Language, 3rd Edition, 1992," the word "debug" is defined as "to search for and eliminate malfunctioning elements or errors in." Consequently, when an error occurs in an

executing program in Song, an exception will be triggered. A handling routine (debug code) is invoked and executed by the processor in order to search for the problem causing the exception and eliminate the error. This process would be considered a debug mode.

31. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Song in view of Grochowski in view of Hennessy, as applied above, and further in view of Matt, as applied above.

32. Referring to claim 13, Song in view of Grochowski in view of Hennessy has taught a method as described in claim 10.

a) Furthermore, Song in view of Grochowski has taught that in a precise watch mode, a request for guard resolution is issued such that an instruction guard is resolved prior to execution of the instruction. Note from column 3, lines 25-35, of Grochowski, that a predicated instruction is prevented from executing until the guard is resolved. This is advantageous in that a time-expensive recovery is not required if the guard is improperly predicted. Therefore, when Song is in precise watch mode, Grochowski has taught that it is beneficial to issue a request to the execution pipeline for guard resolution for the aforementioned reasons. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to issue a request to the execution pipeline for guard resolution during precise watch mode in Song.

b) Song in view of Grochowski in view of Hennessy has not taught selectively causing issue of one of a go command and a debug command responsive to the guard resolution. However, Matt has taught issuing a go command and divert command to the decode unit responsive to receipt of the guard resolution from the execution pipeline, wherein a go command allows the instruction

which caused the breakpoint and subsequent instructions to be normally decoded and executed, and a divert command sets the computer system into a debug mode. See column 4, lines 6-40. More specifically, when a break event occurs, (which would include exceptions taught by Song), the system would issue either a command to resume execution (go command) or a command to execute a service routine (divert command). This allows for real-time execution control for debug functions within a processor, as taught by Matt in column 4, lines 6-9. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Song in view of Grochowski in view of Matt such that an go and divert commands may be issued in order to allow for real-time execution control for debug functions.

33. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Song in view of Grochowski in view of Hennessy, as applied above, and further in view of Adler, as applied above.

34. Referring to claim 15, Song in view of Grochowski has taught a method as described in claim 10. Song in view of Grochowski in view of Hennessy has not taught the specifics set forth in claim 15. However, Adler has taught a system in which the instruction which caused the breakpoint (exception) and subsequent instructions are decoded and executed normally until such time as the guard is resolved wherein, if the guard is resolved such that a position commit signal is generated, the processor is set into a debug mode. See the last 8 lines of the abstract and Fig.9. In essence, predicated instructions are executed speculatively (i.e., executed normally without knowing the actual value of the guard/predicate). When the instruction is at a commit point, and if the instruction's predicate is true, then if that instruction caused a breakpoint (or exception),

then it will be serviced (debug mode). This scheme allows errors that should not have occurred (errors resulting from instructions that should not have executed) to be ignored while servicing the errors that should have occurred (errors resulting from instructions that have properly executed. See column 3, lines 20-42. As a result, it would have been obvious to one of ordinary skill in the art to modify the system taught by Song in view of Grochowski such that it includes the functionality of Adler.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is (703) 746-7239.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

DJH
David J. Huisman
January 12, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100